**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| Escenario de almacenamiento | HashTable | Almacenar tareas y recordatorios en la tabla hash |

**Diseño de Casos de Prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** almacenar tareas | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTable | Public void insert(k key, v value), increaseKey(),maxHeapify  (int i), extractMaximum() | El usuario agrega una tarea en una hash table vacia | Enter ID  Enter title  Enter description  Enter due date  Select type: 1. Task  2. reminder | Tarea almacenada correctamente en la tabla hash, tabla hash Aumenta 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** almacenar tareas | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTable |  | El usuario agrega una tarea y quiere ordenarla por orden de prioridad | Select type: 1. Task  2. reminder  Select if the task is priority: 1. Yes  2. no  If answer is yes select importance level: 3. Very importante  2. important  1. less important | Tareas ordenadas correctamente por elección de prioridad |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** deshacer acción | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTable |  | El usuario agrega una tarea y luego deshace la acción |  |  |